**Exercise 2: Online Bookstore - Creating Basic REST Controllers**

**Book**

package com.bookstore.api.entities;

import lombok.Data;

@Data

public class Book {

private Long id;

private String title;

private String author;

private Double price;

private String isbn;

}

**BookController**

package com.bookstore.api.controllers;

import com.bookstore.api.entities.Book;

import org.springframework.web.bind.annotation.\*;

import java.util.ArrayList;

import java.util.List;

@RestController

@RequestMapping("/books")

public class BookController {

private List<Book> books = new ArrayList<>();

@GetMapping

public List<Book> getAllBooks() {

return books;

}

@GetMapping("/{id}")

public Book getBookById(@PathVariable Long id) {

return books.stream()

.filter(book -> book.getId().equals(id))

.findFirst()

.orElse(null);

}

@PostMapping

public Book createBook(@RequestBody Book book) {

books.add(book);

return book;

}

@PutMapping("/{id}")

public Book updateBook(@PathVariable Long id, @RequestBody Book book) {

Book existingBook = getBookById(id);

if (existingBook != null) {

existingBook.setTitle(book.getTitle());

existingBook.setAuthor(book.getAuthor());

existingBook.setPrice(book.getPrice());

existingBook.setIsbn(book.getIsbn());

return existingBook;

}

return null;

}

@DeleteMapping("/{id}")

public void deleteBook(@PathVariable Long id) {

books.removeIf(book -> book.getId().equals(id));

}

}

In the above code, i have created a BookController class with request mappings for /books. I have implemented methods to handle GET, POST, PUT, and DELETE requests. The controller returns JSON responses and defines the Book entity with attributes like id, title, author, price, and isbn.